

예제로 쉬게 배우는

# 아두이노

김진환 · 장성용 지음

2.1 Arduino에서 컴퓨터로 데이터 전송하기

---

2.2 변수 유형별로 컴퓨터에 전송하기

---

2.3 Arduino에서 시리얼 통신을 이용하여  
데이터 수신하기

---

### UART (Universal Asynchronous Receiver/Transmitter)

RS-232

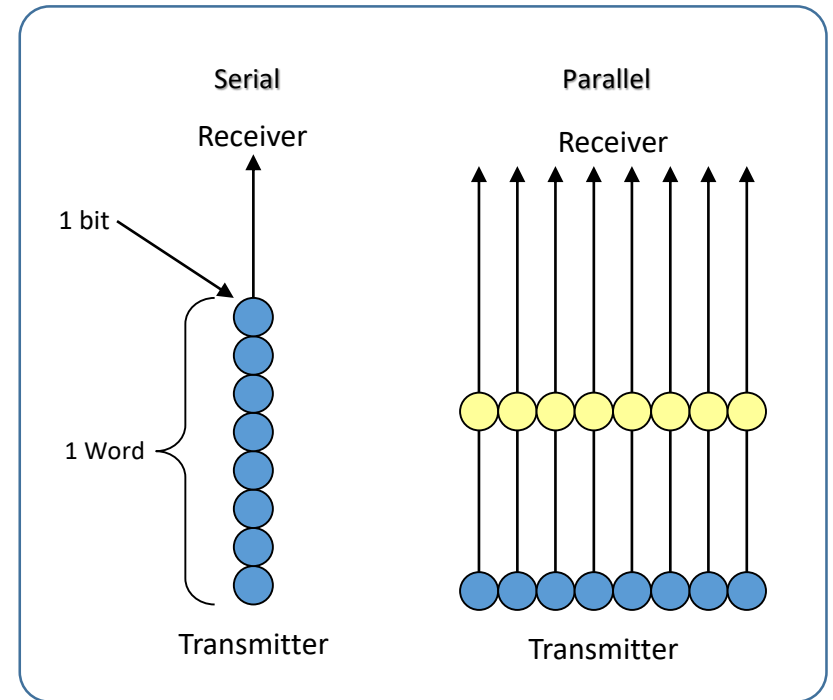
RS-422

RS-485

### Arduino에서는 다음과 같은 목적으로 사용

Debugging : 프로그램의 오류를 수정하는 작업

데이터 통신 : Arduino와 컴퓨터 혹은 다른 장치와의 통신



## 2.1 Arduino에서 컴퓨터로 데이터 전송하기

### EX 2.1 Arduino에서 컴퓨터로 변수와 문자열 전송하기 (1/3)

- 실습목표**
1. Arduino에서 문자열과 데이터를 시리얼 통신을 이용하여 컴퓨터로 전송한다.
  2. 전송할 데이터는 0부터 1초 간격으로 1씩 증가하는 숫자와 'sec'라는 문자열이다.
  2. Arduino IDE의 시리얼 모니터에서 이를 확인해 본다.

**Hardware** Arduino와 PC를 USB 케이블로 연결한다.

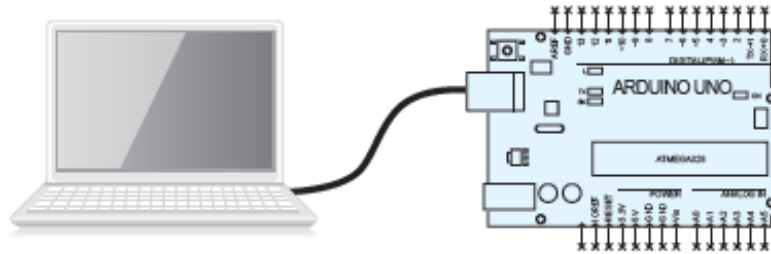


그림 2.1 Arduino와 PC와의 연결

## 2.1 Arduino에서 컴퓨터로 데이터 전송하기

EX 2.1

### Arduino에서 컴퓨터로 변수와 문자열 전송하기 (2/3)

#### Commands

- Serial.begin(전송속도)

시리얼 통신 포트를 컴퓨터와 연결한다. 전송속도는 bps(bits per sec)로 일반적으로 9600으로 설정한다. 19200, 57600, 115200 등의 값을 설정할 수 있다.

- Serial.print(전송내용)

괄호 안의 내용을 시리얼 통신으로 전송한다. 따옴표로 구분된 부분은 텍스트를 직접 전송하고 따옴표 없이 변수를 써주면 변수의 값이 전송된다.

- Serial.println(전송내용)

‘Serial.print’와 같으나 전송 뒤 줄 바꿈을 한다.

- delay(지연시간)

지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 1/1000초 단위로 넣는다. 즉 1초를 지연시키기 위해선 1000의 값을 입력시킨다.

## 2.1 Arduino에서 컴퓨터로 데이터 전송하기

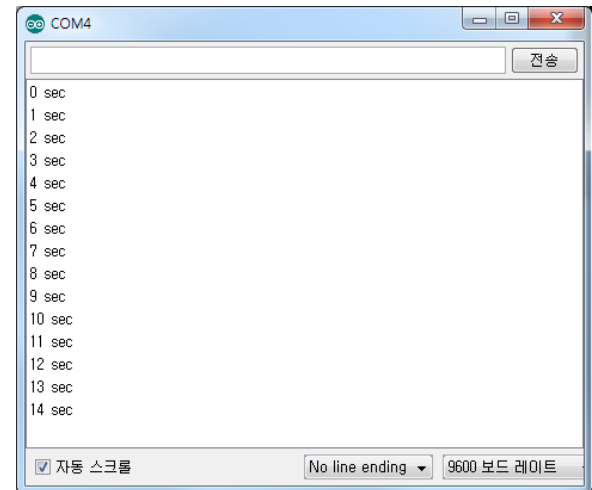
EX 2.1

### Arduino에서 컴퓨터로 변수와 문자열 전송하기 (3/3)

- Sketch 구성**
1. 시리얼통신을 시작한다. 'Serial.begin( )' 명령어로 할 수 있다.
  2. 변수를 전송하기 위해 'Serial.print( )' 명령어를 사용하고,  
문자열 전송 후 줄 바꿈을 하기 위해서 'Serial.println( )' 명령어를 사용한다.
  3. 루프를 1초마다 실행하기 위해서 'delay( )'명령어로 시간지연을 시켜준다.

**실험 결과** IDE의 시리얼 모니터를 실행시켜 Arduino에서 전송되는 메시지를 확인할 수 있다.

- 응용 문제**
1. 2초, 5초 단위로 시간을 변경해 보자.
  2. 자신만의 메시지를 출력해보자.



## 2.2 변수 유형별로 컴퓨터에 전송하기

- ✓ 사용 목적에 따라 다양한 변수 유형 중 선택하여 사용
- ✓ C 언어와 유사함

표 2.1 Arduino 변수 유형

변수 유형	바이트	변수 범위	용도
boolean	1	true(1) 혹은 false(0)	참(1) 아니면 거짓(0) 값을 나타냄.
char	1	-128~127 혹은 아스키 코드 값과 매칭되는 문자	부호가 있는 1바이트 숫자를 나타냄
unsigned char	1	0~255	0~255의 숫자를 나타낼 때 사용함
byte	1	0~255	unsigned char과 동일
int	2	-32,768~32,767	부호가 있는 2바이트 숫자를 나타냄
short	2	-32,768~32,767	int와 동일
unsigned int	2	0~65,535	부호가 없는 2바이트 숫자를 나타냄
word	2	0~65,535	unsigned int와 동일
long	4	-2,147,483,648 ~2,147,483,647	부호가 있는 4바이트 숫자를 나타냄
unsigned long	4	0~4,294,967,295	부호가 없는 4바이트 숫자를 나타냄
float	4	$-3.4028235 \times 10^{38}$ ~ $3.4028235 \times 10^{38}$	소수점 있는 숫자를 나타냄
double <sup>1)</sup>	4 or 8	$-3.4028235 \times 10^{38}$ ~ $3.4028235 \times 10^{38}$ Or $-1.79 \times 10^{308}$ ~ $1.79 \times 10^{308}$	소수점 있는 숫자를 나타냄. 4바이트의 경우 float과 동일 8바이트의 경우 변수 범위가 커짐
String	-	문자열에 따라 다름	문자열을 나타냄.

## 2.2 변수 유형별로 컴퓨터에 전송하기

✓ 실제 전송은 아스키코드 (ASCII Code)를 전송함

ASCII CODE TABLE

10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자
0	0x00	NULL	22	0x16	STN	44	0x2C	,	66	0x42	B	88	0x58	X	110	0x6E	n
1	0x01	SOH	23	0x17	ETB	45	0x2D	-	67	0x43	C	89	0x59	Y	111	0x6F	o
2	0x02	STX	24	0x18	CAN	46	0x2E	.	68	0x44	D	90	0x5A	Z	112	0x70	p
3	0x03	ETX	25	0x19	EM	47	0x2F	/	69	0x45	E	91	0x5B	[	113	0x71	q
4	0x04	EOT	26	0x1A	SUB	48	0x30	0	70	0x46	F	92	0x5C	\	114	0x72	r
5	0x05	ENQ	27	0x1B	ESC	49	0x31	1	71	0x47	G	93	0x5D	]	115	0x73	s
6	0x06	ACK	28	0x1C	FS	50	0x32	2	72	0x48	H	94	0x5E	^	116	0x74	t
7	0x07	BEL	29	0x1D	GS	51	0x33	3	73	0x49	I	95	0x5F	_	117	0x75	u
8	0x08	BS	30	0x1E	RS	52	0x34	4	74	0x4A	J	96	0x60	.	118	0x76	v
9	0x09	HT	31	0x1F	US	53	0x35	5	75	0x4B	K	97	0x61	a	119	0x77	w
10	0x0A	LF	32	0x20	SP	54	0x36	6	76	0x4C	L	98	0x62	b	120	0x78	x
11	0x0B	VT	33	0x21	!	55	0x37	7	77	0x4D	M	99	0x63	c	121	0x79	y
12	0x0C	FF	34	0x22	"	56	0x38	8	78	0x4E	N	100	0x64	d	122	0x7A	z
13	0x0D	CR	35	0x23	#	57	0x39	9	79	0x4F	O	101	0x65	e	123	0x7B	{
14	0x0E	SO	36	0x24	\$	58	0x3A	:	80	0x50	P	102	0x66	f	124	0x7C	
15	0x0F	SI	37	0x25	%	59	0x3B	;	81	0x51	Q	103	0x67	g	125	0x7D	}
16	0x10	DEL	38	0x26	&	60	0x3C	<	82	0x52	R	104	0x68	h	126	0x7E	~
17	0x11	DC1	39	0x27	'	61	0x3D	=	83	0x53	S	105	0x69	i	127	0x7F	DEL
18	0x12	DC2	40	0x28	(	62	0x3E	>	84	0x54	T	106	0x6A	j			
19	0x13	DC3	41	0x29	)	63	0x3F	?	85	0x55	U	107	0x6B	k			
20	0x14	DC4	42	0x2A	*	64	0x40	@	86	0x56	V	108	0x6C	l			
21	0x15	NAK	43	0x2B	+	65	0x41	A	87	0x57	W	109	0x6D	m			

그림 2.2 ASCII 코드표



## 2.2 변수 유형별로 컴퓨터에 전송하기

### EX 2.2 변수 유형별 Arduino에서 컴퓨터로 전송하기 (1/4)

- 실습목표**
1. Arduino에서 컴퓨터로 데이터를 전송할 때 변수 유형별로 출력한다.
  2. char로 선언된 변수, int로 선언된 변수, float로 선언된 변수를 'Serial.print' 명령어를 이용하여 PC로 전송하자.
  3. 'Serial.print' 명령어의 출력 옵션을 변경하여 전송해 보자.
  4. 문자열 변수를 사용해 보자.
  5. 각 변수 유형별 출력되는 차이를 비교해 보자.

**Hardware** Arduino와 PC를 USB 케이블로 연결한다.

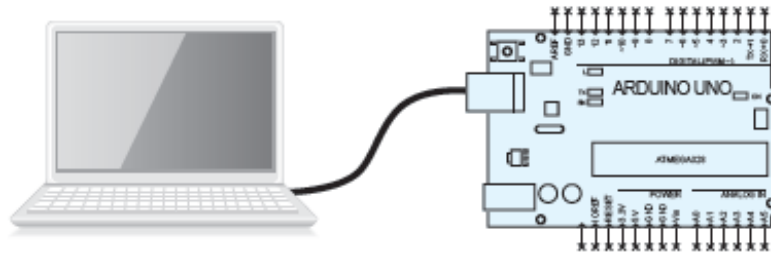


그림 2.1 Arduino와 PC와의 연결

## 2.2 변수 유형별로 컴퓨터에 전송하기

### EX 2.2 변수 유형별 Arduino에서 컴퓨터로 전송하기 (2/4)

- Commands**
- `Serial.write(char 변수);`  
char변수에 해당하는 ASCII 코드값의 문자를 출력한다.
  - `Serial.print(변수,BIN);`  
변수를 2진수(Binary)로 표시한다.
  - `Serial.print(변수,DEC);`  
변수를 10진수(Binary)로 표시한다.
  - `Serial.print(변수,HEX);`  
정해진 변수를 16진수(Hexadecimal)로 표시한다.

## 2.2 변수 유형별로 컴퓨터에 전송하기

EX 2.2

### 변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/4)

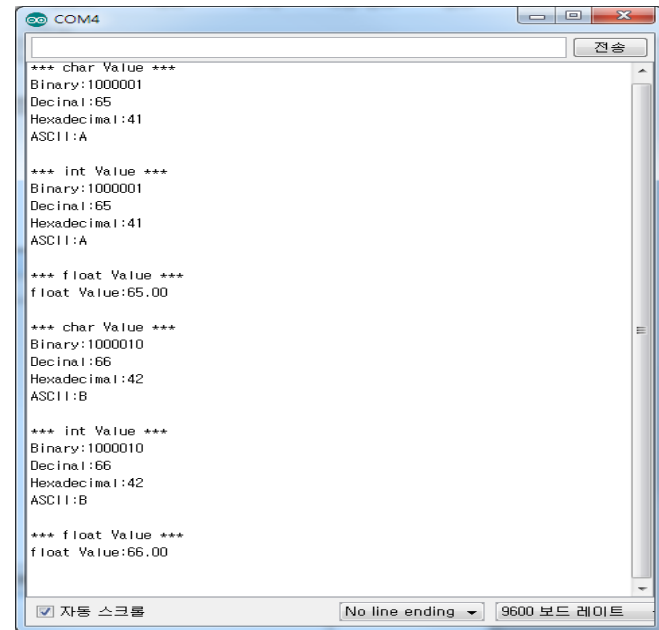
- Sketch 구성**
1. '65'란 숫자를 char형, int형, float형 변수에 각각 저장한다.
  2. 'Binary:', 'Decimal:', 'Hexadecimal:', 'ASCII:' 등 네 가지 문자열을 저장하여 호출하여 사용한다.
  3. 변수에 저장된 숫자를 2진수형, 10진수형, 16진수형, ASCII 코드형 등 Serial.print 명령어의 옵션을 변경하여 전송한다.
  4. 옵션이 변경될 때마다 문자열을 호출하여 함께 출력한다.
  5. loop가 반복될 때마다 숫자를 1씩 증가시킨다. float형은 0.1씩 증가시킨다.

## 2.2 변수 유형별로 컴퓨터에 전송하기

## EX 2.2 변수 유형별 Arduino에서 컴퓨터로 전송하기 (4/4)

**실습 결과** IDE의 시리얼 모니터를 실행시켜 Arduino에서 전송되는 메시지를 확인할 수 있다.

10초 간격으로 증가된 값에 대하여 출력하게 된다. 이때 2진수, 10진수, 16진수로 표시되고, 증가된 숫자에 해당하는 아스키코드의 문자가 전송된다. 'float'형 변수는 소수점이 함께 표시된다.



- 응용 문제**
- 0~15까지 2진수를 10진수와 16진수로 출력하는 스케치를 작성해보자.  
(hint: 10진수 숫자를 16진수 형태로 출력하면 10진수가 16진수로 변경되어 출력된다.)
  - 간단한 연산을 넣어서 결과를 확인해보자.

## 2.3 Arduino에서 시리얼 통신을 이용하여 데이터 수신하기

### EX 2.3 변수 유형별 Arduino에서 컴퓨터로 전송하기 (1/3)

- 실습목표**
1. 컴퓨터에서 Arduino로 0~9의 숫자를 전송한다.
  2. Arduino에서는 전송 받은 숫자만큼 Arduino 보드의 LED를 점멸시킨다.

**Hardware** Arduino와 PC를 USB 케이블로 연결한다.

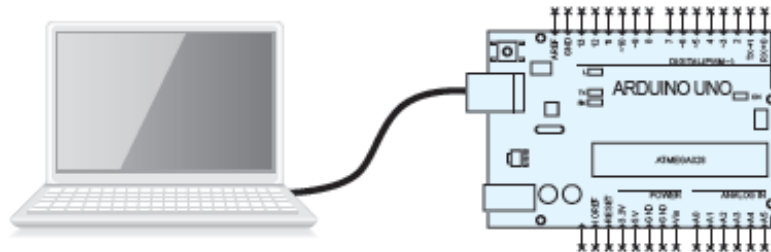


그림 2.1 Arduino와 PC와의 연결

## 2.3 Arduino에서 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

### 변수 유형별 Arduino에서 컴퓨터로 전송하기 (2/3)

#### Commands

- Serial.available( )

시리얼 통신에 수신된 데이터가 있는지 확인한다. 있을 경우 참(true)의 값을 갖는다.

- Serial.read( )

시리얼 통신을 통하여 수신된 값을 읽는다.

- isDigit(변수)

변수의 값이 ASCII 코드의 0~9의 숫자 범위에 있는지 여부를 판단. 범위에 있을 경우 참(true)의 값을 갖는다.

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. '핀번호'에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT\_PULLUP'을 적는다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력(High or Low)을 한다. '핀번호'에는 출력하고자 하는 핀의 번호를, '값'에는 'HIGH' 혹은 'LOW'를 설정하여 High 혹은 Low 출력을 한다.

## 2.3 Arduino에서 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

### 변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/3)

- Sketch 구성**
1. 13번 핀에 연결된 내장 LED를 이용한다.
  2. 시리얼 통신 상태를 감시한 후 시리얼 통신으로 입력되는 데이터가 있을 때 이를 저장한다.
  3. 전송된 값은 ASCII 코드값이므로 이를 숫자로 변경한다.
  4. 숫자만큼 LED를 0.2초 간격으로 점멸시킨다.

**실험 결과** IDE의 시리얼 모니터를 실행시켜 전송란에 0~9의 값을 입력한 후 Arduino의 LED가 입력한 값 만큼 점멸하는지를 확인해 본다..

- 응용 문제**
1. 0~9의 입력 값에 따라 점멸 주기가 변화하는 스케치를 작성해 보자.
  2. 0~9의 숫자를 전송하면 전송된 수의 2진수와 16진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자)

